response to commands generated by the graphics application, and wherein the one or more event generators comprises:

at least one API event generator operatively located in the graphics library, wherein the at least one API event generator performs selected diagnostic operations related to an associated one or more graphics library functions,

wherein the at least one API event generator is configured to be dynamically enabled and disabled to perform the selected diagnostic operations.

24. The hooks module of claim 23, wherein the hooks module further comprises:

a normal operations dispatch table including function pointers to the graphics library functions;

a hooks dispatch table including function pointers to the at least one API event generator; and

a dispatch table manager constructed and arranged to copy selected portions of the normal operations dispatch table and the hooks dispatch table to an active dispatch table in the graphics library, wherein, in response to graphics application function calls, the API calls one of either the at least one API event generator or the graphics library functions in accordance with a function pointer located in the active dispatch table.

25. The hooks module of claim 22, wherein the one or more event generators comprises:

one or more internal event generators adapted to be embedded in the graphics diagnostic library to perform selected diagnostic operations in response to a request by the graphics tool.

26. The hooks module of claim 25, wherein the graphics library further includes graphics hardware control modules for controlling the graphics hardware, and

wherein at least one of the one or more internal event generators is embedded in the graphics hardware control modules to provide the attached graphics diagnostic tool access to an associated operation in the graphics hardware control modules.

27. The hooks module of claim 25, wherein the graphics library includes pipeline control modules for managing a graphics pipeline in the graphics system,

wherein the one or more internal event generators comprise at least one pipeline control module event generator embedded in the pipeline control modules for providing the attached graphics diagnostic tool access to the pipeline control modules.

28. The hooks module of claim 25, wherein the graphics library includes device-specific control modules each configured to manage a specific portion of the graphics hardware, wherein the one or more internal event generators comprise:

device-specific event generators for providing the attached graphics diagnostic tool access to the device-specific control modules.

A1

29. The hooks module of claim 21, wherein the graphics diagnostic tool and the hooks module communicate with each other through a interprocess communications (IPC) mechanisms providing socket communications between the graphics diagnostic tool and the hooks module.

30. The hooks module of claim 21, wherein the one or more event generators include at least one first event generator that causes the graphics system to incur significant performance penalties while performing the diagnostic operations, and at least one second event generator that does not cause the graphics system to incur significant performance penalties while performing the diagnostic operations,

wherein the hooks module dynamically enables the at least one first event generator to temporarily perform the diagnostic operations only when required, and to permanently enable the at least one second event generator to perform the diagnostic operations continually.

31. A method for providing a graphics diagnostic tool access to a computer graphics system having graphics hardware and a graphics library for controlling the graphics hardware in response to function calls received from a graphics application executing thereon, the method comprising the steps of:

(a) determining dynamically during execution of the graphics application and without interrupting the execution of the graphics application whether to install an entry point event generator in the graphics library; and

(b) installing the event generator in the graphics library while the graphics application is executing, wherein the event generator is thereafter responsive to the a graphics application function call.

32. The method of claim 31, wherein the step (b) comprising the steps of:

(1) receiving a graphics library function call issued by the graphics application to invoke a graphics library function;

(2) invoking the event generator associated with that graphics library function;

(3) performing diagnostic operations by the event generator; and

(4) calling the associated graphics library function identified in the function call.

33. The method of claim 31, wherein the graphics library is an OpenGL graphics library.

34. The method of claim 32, wherein the step (b) further comprises the step of:

(5) forwarding results of the selected operations to the graphics tool.

35. The method of claim 32, wherein step (2) comprises the steps of:

i) receiving a request from the graphics tool to perform the diagnostic operation;

ii) installing an active dispatch table in the graphics library wherein the active dispatch table include only function pointers to those API event generators that perform diagnostic operations that will provide results requested by the graphics tool; and

iii) calling one of either the API event generators or the graphics library function in response to a graphics library function call.

36. The method of claim 35, wherein step ii) comprises the steps of:

a) providing a normal operations dispatch table having function pointers to the graphics library functions;

b) providing a hooks dispatch table including function pointers to the API event generators; and

c) copying selected portions of the normal operations dispatch table and the hooks dispatch table to the graphics library to form the active dispatch table in the graphics library.

37. The method of claim 31, further comprises the steps of:

(c) embedding in the graphics library an internal event generator; and

(d) enabling the internal event generators in the graphics library in response to a request from the graphics tool; and

(e) performing, by the internal event generator in response to the graphics tool, selected diagnostic operations, wherein the results provided to the graphics tool includes information pertaining to the graphics system.

A1 Cont.

38. A computer graphics system comprising:

a hooks module integrated within the computer graphics system for dynamically attaching a graphics diagnostic tool to predetermined portions of the computer graphics system while the graphics application is executing and without causing interruption of the execution of the graphics application.

39. The computer graphics system of claim 38, wherein the hooks module comprises:

a plurality of internal event generators embedded in a graphics library of the computer graphics system, each said event generator constructed and arranged to perform diagnostic operations during normal operations of the graphics application in response to requests generated by the graphics diagnostic tool.

40. The computer graphics system of claim 38, wherein at least one of the plurality of internal event generators provides the graphics tool with access to internal state and control flow of predetermined portions of the graphics system, and wherein the at least one internal event generators generate a stream of state information sufficient to recreate a current state of the graphics system.